



ATID Application Development Framework Reference Manual – MSR

Revision: Ver. 0.1

Date: January, 2012

ATID Co., Ltd

Table of Contents

Table of Contents	2
Acronym	3
Revision History	4
1 .NET API Reference	5
1.1 Enumerations	5
1.1.1 MSR_RESULT	5
1.2 Delegates	6
1.2.1 MSRCALLBACK	6
1.3 Methods	6
1.3.1 Open	6
1.3.2 Close	6
1.3.3 IsOpened	6
1.3.4 SetCallback	7
1.3.5 StartReadCard	7
1.3.6 CancelReadCard	8
1.3.7 GetData	8
1.3.8 GetFirmwareVersion	9
2 C/C++ API Reference	10
2.1 Enumerations	10
2.1.1 MSR_RESULT	10
2.2 Constants	11
2.2.1 WM_MSR_RESPONSE	11
2.2.2 MAX_TRACK_BUFFER	11
2.3 Callback function definition	11
2.3.1 MSRCALLBACK	11
2.4 Methods	11
2.4.1 MsrOpen	11
2.4.2 MsrClose	11
2.4.3 MsrIsOpened	12
2.4.4 MsrSetCallback	12
2.4.5 MsrSetHwnd	12
2.4.6 MsrStartReadCard	13
2.4.7 MsrCancelReadCard	13
2.4.8 MsrGetData	14
2.4.9 MsrGetFirmwareVersion	14

Acronym

modules	descriptions
AADF	ATID Application Development Framework
MSR	Magnetic Stripe Reader

Revision History

Version	Date	Reason	Description	Author
0.1	2012/01/17	Draft		Y. J. CHO

1 .NET API Reference

1.1 Enumerations

1.1.1 MSR_RESULT

The calling result of function.

- **MSR_RESULT_SUCCESS**
Success of function execution.
- **MSR_RESULT_INVALID_ARGS**
Invalid parameter.
- **MSR_RESULT_OUTOFMEMORY**
Fail to allocate resources.
- **MSR_RESULT_UNSUPPORTED**
Currently UnsupportedCommands
- **MSR_RESULT_ALREADY_OPENED**
MSR device is already opened.
- **MSR_RESULT_NOT_OPENED**
Calling function without opening.
- **MSR_RESULT_FAILURE**
Fail to execute function.
- **MSR_RESULT_BUSY**
Waiting for card recognition.
- **MSR_RESULT_TIME_OUT**
Cannot recognize card within five seconds(not used)
- **MSR_RESULT_CANCELLED**
카드 인식 대기 상태가 취소됨.
Standby state of card recognition is cancelled.
- **MSR_RESULT_WAITING_END_OF_PRINTING**
Starting to print
- **MSR_RESULT_RECEIVED_END_OF_PRINTING**
Finish printing
- **MSR_RESULT_INVALID_DEVICE**
MSR device is not equipped.

1.2 Delegates

1.2.1 MSRCALLBACK

Callback delegate that is invoked when a data read or data read from the MSR device is canceled. In order to read the card data in application, the callback delegate should be set using SetCallback function. If Callback delegate is executed, the data are processed according to the values of the parameters MSR_RESULT.

Public delegate void **MSRCALLBACK**(MSR_RESULT MsrResult);

1.3 Methods

1.3.1 Open

Allocation system resources and opening MSR device.

MSR_RESULT Open();

Parameters

None

Return Values

MSR_RESULT_SUCCESS will be returned if successfully allocate resources.

1.3.2 Close

Deallocating system resources and closing MSR device.

MSR_RESULT Close();

Parameters

None

Return Values

MSR_RESULT_SUCCESS will be returned if successfully allocate resources.

1.3.3 IsOpened

Checking whether MSR device is open.

BOOL IsOpened ();

Parameters

None

Return Values

Open status of MSR device.

True:Opend, False:Closed

1.3.4 SetCallback

Registering a delegate function that is run when it receives a response(the card is recognized by complete or interrupted) from the MSR device.

MSR_RESULT SetCallback (
 MSRCALLBACK Callback,
);

Parameters

Callback

Delegate function that is executed when a response is received from the MSR device.

Return Values

MSR_RESULT_SUCCESS will be returned if successfully allocate resources.

1.3.5 StartReadCard

Recognizing card Asynchronously.

MSR_RESULT StartReadCard();

Parameters

None

Return Values

MSR_RESULT_SUCCESS will be returned if successfully allocate resources.

Notes

If calling this function and recognizing Card is successful completion, the delegate function is executed, MSR_RESULT_SUCCESS is received to the parameters of the delegate function. When this function is called, card recognition will be waited for permanently until card recognition is successful or standby command is aborted.

If the printer start printing while waiting for the card,

MSR_RESULT_WAITING_END_OF_PRINTING is received to callback delegate and recognizing card is suspend. At this point, the card is ignored, and card recognition will be resumed when the printer output ends and MSR_RESULT_RECEIVED_END_OF_PRINTING is received

1.3.6 CancelReadCard

Stopping recognizing card.

```
MSR_RESULT CancelReadCard ();
```

Parameters

None

Return Values

MSR_RESULT_SUCCESS will be returned if successfully allocate resources.

Notes

If calling this function before StartReadCard is called and card is recognized, the card recognition stops, and MSR_RESULT_CANCELED is received to parameter of delegate function.

1.3.7 GetData

Reading recognized card data.

```
MSR_RESULT GetData (  
    ref string sTrack1,  
    ref string sTrack2,  
    ref string sTrack3,  
    ref string sJIS  
);
```

Parameters

sTrack1

Variables which Track1 information of the card is stored in.

sTrack2

Variables which Track2 information of the card is stored in.

sTrack3

Variables which Track3 information of the card is stored in.

sJIS

Variables which JIS information of the card is stored in.

Return Values

MSR_RESULT_SUCCESS will be returned if successfully allocate resources.

Notes

If the card is recognized completely or is interrupted, the delegate function is called, the application reads the data on the card using this function.

1.3.8 GetFirmwareVersion

Reading firmware version of MSR device.

```
MSR_RESULT GetFirmwareVersion (  
    string sVersion  
);
```

Parameters

sVersion

Variable which firmware version of MSR device will be stored in.

Return Values

MSR_RESULT_SUCCESS will be returned if successfully allocate resources.

2 C/C++ API Reference

2.1 Enumerations

2.1.1 MSR_RESULT

The calling result of function.

- **MSR_RESULT_SUCCESS**
Success of function execution.
- **MSR_RESULT_INVALID_ARGS**
Invalid parameter.
- **MSR_RESULT_OUTOFMEMORY**
Failing to allocate resources.
- **MSR_RESULT_UNSUPPORTED**
Currently Unsupported Commands
- **MSR_RESULT_ALREADY_OPENED**
MSR device is already opened.
- **MSR_RESULT_NOT_OPENED**
Calling function without opening.
- **MSR_RESULT_FAILURE**
Fail to execute function.
- **MSR_RESULT_BUSY**
Waiting for card recognition
- **MSR_RESULT_TIME_OUT**
Cannot recognize card within five seconds(not used)
- **MSR_RESULT_CANCELLED**
Standby state of card recognition is cancelled.
- **MSR_RESULT_WAITING_END_OF_PRINTING**
Starting to print
- **MSR_RESULT_RECEIVED_END_OF_PRINTING**
Finish printing
- **MSR_RESULT_INVALID_DEVICE**
MSR device is not equipped.

2.2 Constants

2.2.1 WM_MSR_RESPONSE

Message that will be transmitted to application when MSR device finish Card recognition or card recognition is cancelled.

- #define **WM_MSR_RESPONSE** WM_USER + 1807

2.2.2 MAX_TRACK_BUFFER

The maximum size of the data on the card.

- #define **MAX_TRACK_BUFFER** 107

2.3 Callback function definition

2.3.1 MSRCALLBACK

Callback delegate that is invoked when a data read or data read from the MSR device is canceled. In order to read the card data in application, the callback delegate should be set using SetCallback function. If Callback delegate is executed, the data are processed according to the values of the parameters MSR_RESULT.

- typedef void (CALLBACK* **MSRCALLBACK**)(MSR_RESULT MsrResult);

2.4 Methods

2.4.1 MsrOpen

Allocating system resources and opening MSR device.

MSR_RESULT MsrOpen();

Parameters

None

Return Values

MSR_RESULT_SUCCESS will be returned if successfully allocate resources.

2.4.2 MsrClose

Deallocating system resources and closing MSR device.

MSR_RESULT MsrClose();

Parameters*None***Return Values**

MSR_RESULT_SUCCESS will be returned if successfully allocate resources.

2.4.3 MsrIsOpened

Checking whether MSR device is open.

```
BOOL MsrIsOpened ();
```

Parameters*None***Return Values**

Open status of MSR device.

True:Open, False:Closed

2.4.4 MsrSetCallback

Registering a delegate function that is run when it receives a response(the card is recognized by complete or interrupted) from the MSR device.

```
MSR_RESULT MsrSetCallback (  
    MSRCALLBACK Callback,  
);
```

Parameters*Callback*

Delegate function that is executed when a response is received from the MSR device.

Return Values

MSR_RESULT_SUCCESS will be returned if successfully allocate resources.

Notes

the data can be processed not registering Callback function but receiving messages to register windows handle of application.

2.4.5 MsrSetHwnd

Registering a windows handle that receives windows messages when receiving a response(the card is recognized by complete or interrupted) from the MSR device.

```
MSR_RESULT MsrSetHwnd (
    HWND hWnd
);
```

Parameters

hWnd

Windows handle of application that will receive messages when receiving response from MSR device.

Return Values

MSR_RESULT_SUCCESS will be returned if successfully allocate resources.

Notes

Data can be processed not registering windows handle but registering callback function.

2.4.6 MsrStartReadCard

Starting to Recognize card Asynchronously.

```
MSR_RESULT MsrStartReadCard();
```

Parameters

None

Return Values

MSR_RESULT_SUCCESS will be returned if successfully allocate resources.

Notes

If calling this function and recognizing Card is successful completion, the delegate function is executed, MSR_RESULT_SUCCESS is received to the parameters of the delegate function. When this function is called, card recognition will be waited for permanently until card recognition is successful or standby command is aborted.

If the printer start printing while waiting for the card,

MSR_RESULT_WAITING_END_OF_PRINTING is received to callback delegate and recognizing card is suspend. At this point, the card is ignored, and card recognition will be resumed when the printer output ends and MSR_RESULT_RECEIVED_END_OF_PRINTING is received

2.4.7 MsrCancelReadCard

Stopping recognizing card.

```
MSR_RESULT MsrCancelReadCard ();
```

Parameters

None

Return Values

MSR_RESULT_SUCCESS will be returned if successfully allocate resources.

Notes

If calling this function before MsrStartReadCard is called and card is recognized, the card recognition stops, and MSR_RESULT_CANCELED is received to parameter of delegate function.

2.4.8 MsrGetData

Reading recognized card data.

```
MSR_RESULT MsrGetData (
    LPWSTR szTrack1,
    LPWSTR szTrack2,
    LPWSTR szTrack3,
    LPWSTR szJIS
);
```

Parameters

szTrack1

Variables which Track1 information of the card is stored in.

szTrack2

Variables which Track2 information of the card is stored in.

szTrack3

Variables which Track3 information of the card is stored in.

szJIS

Variables which JIS information of the card is stored in.

Return Values

MSR_RESULT_SUCCESS will be returned if successfully allocate resources.

Notes

If the card is recognized completely or is interrupted, the delegate function is called, the application reads the data on the card using this function.

2.4.9 MsrGetFirmwareVersion

Reading firmware version of MSR device.

```
MSR_RESULT MsrGetFirmwareVersion (
```

`LPWSTR szVersion`

`);`

Parameters

szVersion

Variable which firmware version of MSR device will be stored in.

Return Values

MSR_RESULT_SUCCESS will be returned if successfully allocate resources.